

Unbiased estimation of second-order parameter sensitivities for stochastic reaction networks

Ankit Gupta and Mustafa Khammash

July 29, 2014

Abstract

This paper deals with the problem of estimating second-order parameter sensitivities for stochastic reaction networks, where the reaction dynamics is modeled as a continuous time Markov chain over a discrete state space. Estimation of such second-order sensitivities (the Hessian) is necessary for implementing the Newton-Raphson scheme for optimization over the parameter space. To perform this estimation, Wolf and Anderson [9] have proposed an efficient finite-difference method, that uses a coupling of perturbed processes to reduce the estimator variance. The aim of this paper is to illustrate that the same coupling can be exploited to derive an exact representation for second-order parameter sensitivity. Furthermore with this representation one can construct an unbiased estimator which is easy to implement. The ideas contained in this paper are extensions of the ideas presented in [4, 5] in the context of first-order parameter sensitivity estimation.

1 Problem Definition

Consider a reaction network with d species and K reactions whose *stoichiometric* vectors are given by ζ_1, \dots, ζ_K . We assume that propensities of various reactions depend on a vector of p parameters $\theta = (\theta_1, \dots, \theta_p)$, which may represent systems's parameters such as, reaction rate constants, hill coefficients, cell volume etc. When the state is x , the k -th reaction fires at rate $\lambda_k(x, \theta)$ and it moves the state to $(x + \zeta_k)$. In the stochastic setting, the reaction dynamics can be represented by a Markov process whose generator is

$$\mathbb{A}_\theta f(x) = \sum_{k=1}^K \lambda_k(x, \theta) \Delta_{\zeta_k} f(x),$$

where $\Delta_{\zeta_k} f(x) = f(x + \zeta_k) - f(x)$.

Let $(X_\theta(t))_{t \geq 0}$ be a process with generator \mathbb{A}_θ . Then its random time change representation (see [2]) is given by

$$X_\theta(t) = X_\theta(0) + \sum_{k=1}^K Y_k \left(\int_0^t \lambda_k(X_\theta(s), \theta) ds \right) \zeta_k,$$

where $\{Y_k : k = 1, \dots, K\}$ is a family of independent unit rate Poisson processes. For any function $f : \mathbb{N}_0^d \rightarrow \mathbb{R}$ that expresses an output of interest, define

$$\Psi_\theta(x, f, t) = \mathbb{E}(f(X_\theta(t)) | X_\theta(0) = x).$$

Then for any $i = 1, \dots, p$, the *first-order* sensitivity

$$S_\theta^{(i)}(x, f, t) = \frac{\partial \Psi_\theta(x, f, t)}{\partial \theta_i} \tag{1.1}$$

measures how sensitive the expected value of the output at time t , $\mathbb{E}(f(X_\theta(t)))$, is to small changes in parameter θ_i . Many methods exist in the literature to estimate $S_\theta^{(i)}(x, f, t)$ (see [7, 8, 1, 4, 5, 6]). In this paper we deal with the problem of estimating the *second-order* sensitivity

$$S_\theta^{(i,j)}(x, f, t) = \frac{\partial^2 \Psi_\theta(x, f, t)}{\partial \theta_i \partial \theta_j},$$

for any $i, j \in \{1, \dots, p\}$. Such a quantity measures the local curvature of the mapping $\theta \mapsto \mathbb{E}(f(X_\theta(t)))$ and this information is useful in implementing optimization schemes such as the Newton Raphson method.

In [9], the authors estimate $S_\theta^{(i,j)}(x, f, t)$ using a finite-difference approximation of the form

$$S_\theta^{(i,j)}(x, f, t) \approx \mathbb{E} \left(\frac{f(X_{\theta_1^\epsilon}(t)) - f(X_{\theta_2^\epsilon}(t)) - f(X_{\theta_3^\epsilon}(t)) + f(X_{\theta_4^\epsilon}(t))}{\epsilon^2} \right), \quad (1.2)$$

for a small ϵ , where each X_{θ^ϵ} is a process with generator $\mathbb{A}_{\theta^\epsilon}$ and initial state x , and θ_l^ϵ -s denote perturbations of the parameter θ defined as follows:¹

$$\theta_1^\epsilon = \theta + (e_i + e_j)\epsilon, \quad \theta_2^\epsilon = \theta + e_i\epsilon, \quad \theta_3^\epsilon = \theta + e_j\epsilon \quad \text{and} \quad \theta_4^\epsilon = \theta.$$

Moreover the processes $X_{\theta_1^\epsilon}, X_{\theta_2^\epsilon}, X_{\theta_3^\epsilon}$ and $X_{\theta_4^\epsilon}$, are intelligently coupled to lower the variance of the associated estimator. The main drawback of finite-difference approximation is that it introduces a *bias* in the estimate and generally the size or even the sign of this bias is unknown, which can cause problems in certain applications.

Interestingly, the coupling described in [9] can be used to derive an exact formula for $S_\theta^{(i,j)}(x, f, t)$ by extending the ideas presented in [4] in the context of first-order sensitivity. The advantage of such a formula is that it allows one to construct an efficient unbiased estimator for $S_\theta^{(i,j)}(x, f, t)$, in the same way as the formula for the first-order sensitivity $S_\theta^{(i)}(x, f, t)$ in [4] was used in [5] to devise an unbiased estimator for this quantity. In Section 2 we present our main result that expresses $S_\theta^{(i,j)}(x, f, t)$ as the expectation of a certain random variable. In Section 3 we describe how this result can be used for obtaining unbiased estimates of $S_\theta^{(i,j)}(x, f, t)$ in an efficient way.

2 Main Result

From now on, let $\lambda_0(x, \theta)$ be the function denoting the sum of propensities

$$\lambda_0(x, \theta) = \sum_{k=1}^K \lambda_k(x, \theta).$$

Theorem 2.1 Suppose $(X_\theta(t))_{t \geq 0}$ is the Markov process with generator \mathbb{A}_θ and initial state x_0 . Let σ_l be the l -th jump time of the process for $l = 0, 1, \dots$. Then for any function $f : \mathbb{N}_0^d \rightarrow \mathbb{R}$ and $t \geq 0$, $S_\theta^{(i,j)}(x_0, f, t) = \mathbb{E} \left(s_\theta^{(i,j)}(x_0, f, t) \right)$ where

$$\begin{aligned} s_\theta^{(i,j)}(x_0, f, t) &= \sum_{k=1}^K \left[\int_0^t \frac{\partial^2 \lambda_k(X_\theta(s), \theta)}{\partial \theta_i \partial \theta_j} \Delta_{\zeta_k} f(X_\theta(s)) ds \right. \\ &+ \sum_{l=0, \sigma_l < t} \frac{\partial \lambda_k(X_\theta(\sigma_l), \theta)}{\partial \theta_i} \left(\int_0^{t-\sigma_l} (S_\theta^{(j)}(X_\theta(\sigma_l) + \zeta_k, f, t - \sigma_l - s) - S_\theta^{(j)}(X_\theta(\sigma_l), f, t - \sigma_l - s)) e^{-\lambda_0(X_\theta(\sigma_l), \theta)s} ds \right) \\ &+ \sum_{l=0, \sigma_l < t} \frac{\partial \lambda_k(X_\theta(\sigma_l), \theta)}{\partial \theta_j} \left(\int_0^{t-\sigma_l} (S_\theta^{(i)}(X_\theta(\sigma_l) + \zeta_k, f, t - \sigma_l - s) - S_\theta^{(i)}(X_\theta(\sigma_l), f, t - \sigma_l - s)) e^{-\lambda_0(X_\theta(\sigma_l), \theta)s} ds \right) \end{aligned} \quad (2.3)$$

¹Here e_i denotes the vector $(0, \dots, 0, 1, 0, \dots) \in \mathbb{R}^q$ where the 1 is at the i -th location.

$$+ \sum_{l=0, \sigma_l < t}^{\infty} \frac{\partial^2 \lambda_k(X_{\theta}(\sigma_l), \theta)}{\partial \theta_i \partial \theta_j} \left(\int_0^{t-\sigma_l} (\Psi_{\theta}(X_{\theta}(\sigma_l) + \zeta_k, f, t - \sigma_l - s) - \Psi_{\theta}(X_{\theta}(\sigma_l), f, t - \sigma_l - s) - \Delta_{\zeta_k} f(X_{\theta}(\sigma_l))) e^{-\lambda_0(X_{\theta}(\sigma_l), \theta)s} ds \right) \Big],$$

Proof. The proof follows by a simple extension of the ideas presented in [4]. We start with the finite-difference approximation of the form (1.2), where the processes $X_{\theta_1^{\epsilon}}, X_{\theta_2^{\epsilon}}, X_{\theta_3^{\epsilon}}$ and $X_{\theta_4^{\epsilon}}$ are coupled in the same way as described in [9]. Using Dynkin's formula and exploiting the coupling, we can pass to the limit $\epsilon \rightarrow 0$ and prove the relation given above. The details of the proof shall be provided elsewhere. \square

Observe that Theorem 2.1 expresses the second-order sensitivity as the expectation of a random variable which only involves first-order sensitivities and expectations of the underlying Markov process. Since many efficient methods exist to estimate such first-order sensitivities and expectations, one can hope to use Theorem 2.1 to construct an efficient estimator for the second-order sensitivity. However the main difficulty is that one has to estimate a “new” quantity (first-order sensitivity and/or expectation) at each jump time in the observation time period $[0, t]$. This can be very cumbersome as the number of jumps can be very high. However using the ideas in [5] we can get around this problem and only estimate these “new” quantities at a *small* number of jump times, and still achieve an unbiased estimate for the second-order sensitivity. We describe this approach in the next section.

3 Algorithm

We use the same notation as in Theorem 2.1. Let $(X_{\theta}(t))_{t \geq 0}$ be the Markov process with generator \mathbb{A}_{θ} , initial state x_0 , and let σ_l be the l -th jump time of the process for $l = 0, 1, \dots$. The total number of jumps until time t is given by the random variable

$$\eta_t = \max\{i \geq 0 : \sigma_i < t\}. \quad (3.4)$$

For simplicity we assume that there are no absorbing states (that is, $\lambda_0(x, \theta) > 0$ for all $x \in \mathbb{N}_0^d$). For each $l = 0, \dots, \eta_t$ let γ_l be an independent exponentially distributed random variable with rate $\lambda_0(X_{\theta}(\sigma_l), \theta)$ and define

$$\Gamma_l = \begin{cases} 1 & \text{if } \gamma_l < (t - \sigma_l) \\ 0 & \text{otherwise} \end{cases}. \quad (3.5)$$

For each $k \in \{1, \dots, K\}$ and $q \in \{i, j\}$ let $\beta_{kl}^{(q)}$ be given by

$$\beta_{kl}^{(q)} = \text{Sign} \left(\frac{\partial \lambda_k(X_{\theta}(\sigma_l), \theta)}{\partial \theta_q} \right) \quad \text{where} \quad \text{Sign}(x) = \begin{cases} 1 & \text{if } x > 0 \\ -1 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \end{cases}.$$

Similarly let

$$\beta_{kl}^{(i,j)} = \text{Sign} \left(\frac{\partial^2 \lambda_k(X_{\theta}(\sigma_l), \theta)}{\partial \theta_i \partial \theta_j} \right).$$

Now we choose a normalizing constant $c > 0$, which specifies the “density” of jump times at which we estimate a new quantity of the form $S_{\theta}^{(i)}(\cdot), S_{\theta}^{(j)}(\cdot)$ or $\Psi_{\theta}(\cdot)$. More details on the role of c and how it can be chosen can be found in [5]. Please note that the estimator we construct will remain unbiased for any choice of c , but its variance may vary. For each $q \in \{i, j\}$, if $\beta_{kl}^{(q)} \neq 0$ and $\Gamma_l = 1$, then let $\rho_{kl}^{(q)}$ be an independent \mathbb{N}_0 -valued random variable whose distribution is Poisson with parameter

$$\frac{c}{\lambda_0(X_{\theta}(\sigma_l), \theta)} \left| \frac{\partial \lambda_k(X_{\theta}(\sigma_l), \theta)}{\partial \theta_q} \right|. \quad (3.6)$$

Similarly if $\beta_{kl}^{(i,j)} \neq 0$ and $\Gamma_l = 1$, then let $\rho_{kl}^{(i,j)}$ be an independent \mathbb{N}_0 -valued random variable whose distribution is Poisson with parameter

$$\frac{c}{\lambda_0(X_\theta(\sigma_l), \theta)} \left| \frac{\partial^2 \lambda_k(X_\theta(\sigma_l), \theta)}{\partial \theta_i \partial \theta_j} \right|. \quad (3.7)$$

Let

$$\Delta t_l = \begin{cases} (\sigma_{l+1} - \sigma_l) & \text{for } l = 0, \dots, \eta_t - 1 \\ (T - \sigma_\eta) & \text{for } l = \eta_t \end{cases}$$

and define

$$\begin{aligned} \hat{s}_\theta^{(i,j)}(x_0, f, t) = & \sum_{k=1}^K \sum_{l=0}^{\eta_t} \left[\frac{\partial^2 \lambda_k(X_\theta(\sigma_l), \theta)}{\partial \theta_i \partial \theta_j} \Delta_{\zeta_k} f(X_\theta(\sigma_l)) \left(\Delta t_l - \frac{\Gamma_l}{\lambda_0(X_\theta(\sigma_l), \theta)} \right) \right. \\ & \left. + \frac{1}{c} \Gamma_l \left(\beta_{kl}^{(i)} \rho_{kl}^{(i)} \hat{S}_{kl}^{(i)} + \beta_{kl}^{(j)} \rho_{kl}^{(j)} \hat{S}_{kl}^{(j)} + \beta_{kl}^{(i,j)} \rho_{kl}^{(i,j)} \hat{D}_{kl} \right) \right], \end{aligned} \quad (3.8)$$

where the construction of random variables $\hat{S}_{kl}^{(i)}$, $\hat{S}_{kl}^{(j)}$ and \hat{D}_{kl} is described below.

Let $(Z_1(t))_{t \geq 0}$ and $(Z_2(t))_{t \geq 0}$ be two processes with random time change representations given by:

$$\begin{aligned} Z_1(t) = & (X_\theta(\sigma_l) + \zeta_k) + \sum_{k=1}^K \hat{Y}_k \left(\int_0^t \lambda_k(Z_1(s), \theta) \wedge \lambda_k(Z_2(s), \theta) ds \right) \zeta_k \\ & + \sum_{k=1}^K \hat{Y}_k^{(1)} \left(\int_0^t (\lambda_k(Z_1(s), \theta) - \lambda_k(Z_1(s), \theta) \wedge \lambda_k(Z_2(s), \theta)) ds \right) \zeta_k \\ \text{and } Z_2(t) = & X_\theta(\sigma_l) + \sum_{k=1}^K \hat{Y}_k \left(\int_0^t \lambda_k(Z_1(s), \theta) \wedge \lambda_k(Z_2(s), \theta) ds \right) \zeta_k \\ & + \sum_{k=1}^K \hat{Y}_k^{(2)} \left(\int_0^t (\lambda_k(Z_2(s), \theta) - \lambda_k(Z_1(s), \theta) \wedge \lambda_k(Z_2(s), \theta)) ds \right) \zeta_k, \end{aligned}$$

where $\{\hat{Y}_k, \hat{Y}_k^{(1)}, \hat{Y}_k^{(2)} : k = 1, \dots, K\}$ is an independent family of unit rate Poisson processes. Let

$$\hat{D}_{kl} = f(Z_1(t - \sigma_l - \gamma_l)) - f(Z_2(t - \sigma_l - \gamma_l)), \quad (3.9)$$

then we must have

$$\mathbb{E} \left(\hat{D}_{kl} | \mathcal{F}_t \right) = \Psi_\theta(X_\theta(\sigma_l) + \zeta_k, f, t - \sigma_l - \gamma_l) - \Psi_\theta(X_\theta(\sigma_l), f, t - \sigma_l - \gamma_l), \quad (3.10)$$

where $\{\mathcal{F}_s\}_{s \geq 0}$ is the filtration generated by the process $(X_\theta(t))_{t \geq 0}$.

We now construct $\hat{S}_{kl}^{(q)}$ for each $q \in \{i, j\}$. Using the *Poisson Path Algorithm* (PPA) given in [5], with the underlying Markov process as Z_1 , we can generate one realization of a random variable $\hat{s}_1^{(q)}$ whose expectation is

$$\mathbb{E} \left(\hat{s}_1^{(q)} \right) = S_\theta^{(q)}(X_\theta(\sigma_l) + \zeta_k, f, t - \sigma_l - \gamma_l).$$

Similarly using PPA with the underlying Markov process as Z_2 , we can generate one realization of another random variable $\hat{s}_2^{(q)}$ whose expectation is

$$\mathbb{E} \left(\hat{s}_2^{(q)} \right) = S_\theta^{(q)}(X_\theta(\sigma_l), f, t - \sigma_l - \gamma_l).$$

Defining

$$\hat{S}_{kl}^{(q)} = \hat{s}_1^{(q)} - \hat{s}_2^{(q)}, \quad (3.11)$$

we must have that

$$\mathbb{E} \left(\widehat{S}_{kl}^{(q)} | \mathcal{F}_t \right) = S_{\theta}^{(q)}(X_{\theta}(\sigma_l) + \zeta_k, f, t - \sigma_l - \gamma_l) - S_{\theta}^{(q)}(X_{\theta}(\sigma_l), f, t - \sigma_l - \gamma_l). \quad (3.12)$$

Using relations (3.10) and (3.12) one can show using a simple conditioning argument that

$$\mathbb{E} \left(s_{\theta}^{(i,j)}(x_0, f, t) \right) = \mathbb{E} \left(\widehat{s}_{\theta}^{(i,j)}(x_0, f, t) \right),$$

where $s_{\theta}^{(i,j)}(x_0, f, t)$ and $\widehat{s}_{\theta}^{(i,j)}(x_0, f, t)$ are defined by (2.3) and (3.8) respectively. Hence Theorem 2.1 guarantees that by generating realizations of the random variable $\widehat{s}_{\theta}^{(i,j)}(x_0, f, t)$, we can obtain an unbiased estimate for second-order parameter sensitivity $S_{\theta}^{(i,j)}(x_0, f, t)$.

A single realization of the random variable $\widehat{s}_{\theta}^{(i,j)}(x_0, f, t)$ (given by (3.8)) can be computed using GENERATESAMPLE(x_0, t, c) (Algorithm 2). This method simulates the process X_{θ} according to SSA and at each state x and jump time $s = \sigma_l$, the following happens:

- The exponential random variable γ (where $\gamma = \gamma_l$ in (3.8)) is generated and the corresponding Γ_l (see (3.5)) is calculated.
- If $\gamma < (t-s)$ then for each $k = 1, \dots, K$ such that either $\partial \lambda_k(x, \theta) / \partial \theta_i$, $\partial \lambda_k(x, \theta) / \partial \theta_j$ or $\partial^2 \lambda_k(x, \theta) / \partial \theta_i \partial \theta_j$ is non-zero, we generate the appropriate Poisson random variable ($\rho_{kl}^{(i)}, \rho_{kl}^{(j)}$ or $\rho_{kl}^{(i,j)}$), and if this random variable is positive then the appropriate quantity ($\widehat{S}_{kl}^{(i)}, \widehat{S}_{kl}^{(j)}$ or \widehat{D}_{kl}) is calculated and the sample value is updated according to (3.8).

In this algorithm, we assume that the function *rand()* returns independent samples from the uniform distribution on $[0, 1]$. Moreover $n \sim \text{POISSON}(r)$ implies that n is an independently generated random variable having Poisson distribution with parameter r . When the state of the process is x , the next time increment (Δs) and reaction index (k), as prescribed by Gillespie's SSA [3], can be calculated using function SSA(x) (see Algorithm 1).

Algorithm 1 Computes the next time increment (Δs) and reaction index (k) for Gillespie's SSA

```

1: function SSA( $x$ )
2:   Set  $r_1 = \text{rand}()$  ,  $r_2 = \text{rand}()$  and  $k = 0$ 
3:   Calculate  $\Delta s = -\log(r_1) / \lambda_0(x, \theta)$ 
4:   Set  $S = 0$ 
5:   while  $S < r_2$  do
6:     Update  $k \leftarrow k + 1$ 
7:     Update  $S \leftarrow S + \lambda_k(x, \theta) / \lambda_0(x, \theta)$ 
8:   end while
9:   return ( $\Delta s, k$ )
10: end function

```

References

- [1] D. Anderson. An efficient finite difference method for parameter sensitivities of continuous time markov chains. *SIAM: Journal on Numerical Analysis*, 2012. 1
- [2] S. N. Ethier and T. G. Kurtz. *Markov processes*. Wiley Series in Probability and Mathematical Statistics: Probability and Mathematical Statistics. John Wiley & Sons Inc., New York, 1986. Characterization and convergence. 1
- [3] D. T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, 1977. 3

- [4] A. Gupta and M. Khammash. Unbiased estimation of parameter sensitivities for stochastic chemical reaction networks. *SIAM : Journal on Scientific Computing*, 35(6):2598 – 2620, 2013. (document), 1, 1, 2
- [5] A. Gupta and M. Khammash. An efficient and unbiased method for sensitivity analysis of stochastic reaction networks. *Unpublished. Available on arXiv: 1402.3076*, 2014. (document), 1, 1, 2, 3, 3
- [6] S. Plyasunov and A. Arkin. Efficient stochastic sensitivity analysis of discrete event systems. *Journal of Computational Physics*, 221:724–738, 2007. 1
- [7] M. Rathinam, P. W. Sheppard, and M. Khammash. Efficient computation of parameter sensitivities of discrete stochastic chemical reaction networks. *Journal of Chemical Physics*, 132, 2010. 1
- [8] P. W. Sheppard, M. Rathinam, and M. Khammash. A pathwise derivative approach to the computation of parameter sensitivities in discrete stochastic chemical systems. *Journal of Chemical Physics*, 136, 2012. 1
- [9] E. S. Wolf and D. F. Anderson. A finite difference method for estimating second order parameter sensitivities of discrete stochastic chemical reaction networks. *The Journal of Chemical Physics*, 137(22), 2012. (document), 1, 1, 2

Algorithm 2 Generates one realization of $\hat{s}_{\theta}^{(i,j)}(x_0, f, t)$ according to (3.8)

```

1: function GENERATESAMPLE( $x_0, t, c$ )
2:   Set  $x = x_0, s = 0$  and  $S = 0$ 
3:   while  $s < t$  do
4:     Calculate  $(\Delta s, k_0) = \text{SSA}(x)$ 
5:     Update  $\Delta s \leftarrow \min\{\Delta s, t - s\}$  and set  $\gamma = -\frac{\log(\text{rand}())}{\lambda_0(x, \theta)}$ 
6:     if  $\gamma \geq (t - s)$  then
7:       Update  $S \leftarrow S + \left(\frac{\partial \lambda_k^2(x, \theta)}{\partial \theta_i \partial \theta_j}\right) (f(x + \zeta_k) - f(x)) \Delta s$ 
8:     else
9:       Update  $S \leftarrow S + \left(\frac{\partial \lambda_k^2(x, \theta)}{\partial \theta_i \partial \theta_j}\right) (f(x + \zeta_k) - f(x)) \left(\Delta s - \frac{1}{\lambda_0(x, \theta)}\right)$ 
10:      for  $k = 1$  to  $K$  do
11:        Set  $r^{(i)} = \left|\frac{\partial \lambda_k(x, \theta)}{\partial \theta_i}\right|$ ,  $r^{(j)} = \left|\frac{\partial \lambda_k(x, \theta)}{\partial \theta_j}\right|$  and  $r^{(i,j)} = \left|\frac{\partial^2 \lambda_k(x, \theta)}{\partial \theta_i \partial \theta_j}\right|$ 
12:        Set  $\beta^{(i)} = \text{Sign}\left(\frac{\partial \lambda_k(x, \theta)}{\partial \theta_i}\right)$ ,  $\beta^{(j)} = \text{Sign}\left(\frac{\partial \lambda_k(x, \theta)}{\partial \theta_j}\right)$  and  $\beta^{(i,j)} = \text{Sign}\left(\frac{\partial^2 \lambda_k(x, \theta)}{\partial \theta_i \partial \theta_j}\right)$ 
13:        if  $r^{(i)} > 0$  then
14:          Set  $n \sim \text{POISSON}\left(\frac{r^{(i)}c}{\lambda_0(x, \theta)}\right)$ 
15:          if  $n > 0$  then
16:            Calculate  $\hat{S}_{kl}^{(i)}$  according to (3.11) with  $q = i, \sigma_l = s, \gamma_l = \gamma$  and  $X_{\theta}(\sigma_l) = x$ .
17:            Update  $S \leftarrow S + \left(\frac{\beta^{(i)}n}{c}\right) \hat{S}_{kl}^{(i)}$ 
18:          end if
19:        end if
20:        if  $r^{(j)} > 0$  then
21:          Set  $n \sim \text{POISSON}\left(\frac{r^{(j)}c}{\lambda_0(x, \theta)}\right)$ 
22:          if  $n > 0$  then
23:            Calculate  $\hat{S}_{kl}^{(j)}$  according to (3.11) with  $q = j, \sigma_l = s, \gamma_l = \gamma$  and  $X_{\theta}(\sigma_l) = x$ .
24:            Update  $S \leftarrow S + \left(\frac{\beta^{(j)}n}{c}\right) \hat{S}_{kl}^{(j)}$ 
25:          end if
26:        end if
27:        if  $r^{(i,j)} > 0$  then
28:          Set  $n \sim \text{POISSON}\left(\frac{r^{(i,j)}c}{\lambda_0(x, \theta)}\right)$ 
29:          if  $n > 0$  then
30:            Calculate  $\hat{D}_{kl}$  according to (3.9) with  $\sigma_l = s, \gamma_l = \gamma$  and  $X_{\theta}(\sigma_l) = x$ .
31:            Update  $S \leftarrow S + \left(\frac{\beta^{(i,j)}n}{c}\right) \hat{D}_{kl}$ 
32:          end if
33:        end if
34:      end for
35:    end if
36:    Update  $s \leftarrow s + \Delta s$  and  $x \leftarrow x + \zeta_{k_0}$ 
37:  end while
38:  return  $S$ 
39: end function

```
